

Using OWA for Persian Part of Speech Tagging

Abolfazl Aleahmad^a, Yoosef Ramezani^a, Farhad Oroumchian^b

^aElectrical and Computer Engineering Department, University of Tehran

^bCollege of IT, University of Wollongong in Dubai

{a.aleahmad ,y.ramezani}@ece.ut.ac.ir, FarhadO@uow.edu.au

ABSTRACT

Part of Speech (POS) tagging is an essential part of text processing applications. A POS tagging system assigns a tag to each word of its input text specifying its grammatical properties. Several part of speech tagging systems were developed on the well known BIJANKHAN Farsi tagged corpus that contains 2500000+ tokens and each of them has different precision. On the other hand as OWA (Ordered Weighted Averaging) is a famous method in data fusion domain that also can be used for decision making, we applied this method to fuse the final result of the POS tagging systems. In this study we used OWA method to fuse the result of three different POS tagging systems, namely MLE (Maximum Likelihood Estimation), TnT tagger and PTT (Persian Tree Tagger). We ran the three methods on the BIJANKHAN corpus then applied OWA method to fuse their results. Our results show that using fusion techniques can improve the results of MLE and PTT tagging systems.

1. INTRODUCTION

Part-of-speech tagging selects the most likely sequence of syntactic categories for the words in a sentence. It determines grammatical characteristics of the words, such as part of speech, grammatical number, gender, person, etc. This task is not trivial since many words are ambiguous: for example, English word "fly" can be a noun (e.g. a fly is a small insect) or a verb (e.g. the birds will fly north in summer). Such phenomenon is known practically in most of the languages. Although there are many models and implementations available for the task of tagging, most of them are designed for and tested on English texts; less work has been done on tagging and tagger evaluation for languages like Persian that have quite different properties and script. There are many different models for tagging which differ on their internal model or the amount of training or intervention of information they need. So it seems reasonable to combine the result of different tagging models to produce better results.

Some of our colleagues in linguistics laboratory of university of Tehran have implemented different POS tagging models on a Persian corpus named BIJANKHAN [1] and each of the models has its own characteristics and produced different precisions. Some of them are MBT (Memory Based Tagger), MLE (Maximum Likelihood Estimation) [2], TnT tagger [3] and PTT (Persian Tree Tagger) [4]. On the other hand there are many different information fusion techniques reported in literature that can be used to fuse different systems in feature level, data level or decision level. One of them is

OWA (Ordered Weighted Averaging) that is a famous method in data fusion domain and can also be used for decision making [5].

In this study we used OWA method to fuse the result of three different POS tagging systems, namely MLE, PTT and TnT. The results of our fusion experiments on the Persian tagger systems show that using fusion techniques, we can produce better results than MLE and PTT. The rest of this paper is organized as follows:

Section 2 describes the BIJANKHAN corpus that we used in our experiments, section 3 introduces the OWA method and the MLE, TnT and PTT systems, in section 4 and 5 we will describe our experimentation and its results and at last we will conclude our study.

2. THE BIJANKHAN CORPUS

The corpus which was used in this research is a part of the BIJANKHAN's tagged Persian corpus [6], which is maintained in the linguistics laboratory of the University of Tehran.

This corpus is gathered from daily news and common texts and contains more than 2500000 manually tagged words. In our experiments we split the corpus into two parts containing 80% and 20% of the tokens to be used as training and test parts of our experiments, respectively. The number of tokens in training and test set are shown in the table below:

Table 1. Number of tokens in training and test set

Training Set (80%)	Test Set (20%)	Total
2078595	519621	2598216

More description on the BIJANKHAN corpus like Statistical analysis of the corpus and tags distribution can be found in [1]. But here we will describe the BIJANKHAN corpus at glance.

In BIJANKHAN corpus the tag "N_SING" (Noun-Singular) is the most common tag in both training and test sets (totally 967546 times in corpus). On the other hand, the "NN" (Number) tag has appeared only twice in the training set and never in the test set. Interestingly only the tags "ADJ_SIM" (Simple-Adjective), "CON" (Conjunction), "DELM" (Delimiter), "N_PL" (Noun-Plural), "N_SING" and "P" (Preposition) has appeared more than 6% in both training and test sets. Generally, the percentages of the occurrence are very similar in both test and training sets. This means that the test sets and training sets have similar characteristics and the result obtained through experiments are reliable indicators for behavior of the algorithms.

Another interesting characteristic about the corpus is the ambiguity of the words. For instance, the word "آسمان" which means "the sky" in English is always tagged with "N_SING" in the whole corpus; but a word like "بالا" which means "high or above" has been tagged by several tags ("ADJ_SIM", "ADV", "ADV_NI", "N_SING", "P", and "PRO"). Table 2 reports the number of words with various numbers of tags. It shows that majority of the words in the corpus have only a single POS tag (91%) and only a small fraction of the words have been assigned more than 3 different POS tags.

Table 2. Number of Different Tags

Number of words	Number of different tags
62568	1
4838	2
493	3
118	4
49	5
14	6
7	7
2	8
2	9
1	10

3. RELATED WORKS

We applied information fusion on MLE, PTT and TnT to produce better results. So, in this part first we will introduce OWA method and then we will review the previous POS tagging works that was done on Persian language.

3.1. Ordered Weighted Averaging

The notion of OWA operators was first introduced in [5] regarding the problem of aggregating multi-criteria to form an overall decision function. A mapping

$$F : [0,1]^n \rightarrow [0,1]$$

is called an OWA operator of dimension n if it is associated with a weighting vector $W = [w_1, \dots, w_n]$, such that 1) $w_i \in [0; 1]$ and 2) $\sum_i w_i = 1$, and

$$F(a_1, \dots, a_n) = \sum_{i=1}^n w_i b_i$$

Where b_i is the i -th largest element in the collection a_1, \dots, a_n . OWA operators provide a type of aggregation operators which lay between the "and" and the "or" aggregation. As suggested by Yager [5], there exist at least two methods for obtaining weights w_i 's. The first approach is to use some kind of learning mechanism. That is, we use some sample data, arguments and associated aggregated values and try to fit the weights to this collection of sample data. The second approach is to give some semantics or meaning to the weights. Then, based on these semantics we can directly provide the values for the weights.

3.2. Maximum Likelihood Estimation

In this approach, for every word in the training set we calculated the tag which is assigned to the word more than the other tags [7]. For this purpose, we calculated the maximum likelihood probabilities for each tag assigned to any word in the training set. Then we pick the tag with the greater maximum likelihood probability for each word and

make it the only tag assignable to that word. We call this tag the *designated* tag for that word. In order to evaluate this method we analyze the words in the test set and assign the *designated* tags to the words in the test set.

H.Amiri, et al. have used this method on the BIJANKHAN collection but they also used some post processing techniques to improve their results accuracy, Because the MLE method doesn't have an acceptable accuracy for unknown words [2]. As an example, in Persian language plural nouns have morphemes like "ها", "های", "ان", "ات" and etc. at their tail. For example the word "نیمکت" (bench in English) is a single noun ("N_SING") and "نیمکت ها" (benches in English) is plural noun ("N_PL"). So they used this fact to improve their results.

3.3. TnT Tagger

In recent years, there has been a growing interest in data-driven machine-learning disambiguation methods, which can be used in many situations such as tagging. Among the most promising disambiguation methods are those based on learning decision list [8] which is an ordered list of conjunctive rules. The decision list induction problem is to identify from a training set of examples the decision list that will most accurately classify future examples.

TnT tagger is proposed by Thorsten Brants and in literature its efficiency is reported to be as one of the best and fastest on different languages such as German, English, Slovene, and Spanish [9]. Brants's TnT (Trigrams'n'Tags) tagger is a statistical part of speech tagger, trainable on different languages and virtually any tag set. The component for parameter generation is trained on a tagged corpus. The system incorporates several methods of smoothing and of handling unknown words. TnT is not optimized for a particular language; instead, it is optimized for training on a large variety of corpora. The tagger is an implementation of the Viterbi algorithm for second orders Markov models. The main paradigm used for smoothing is linear interpolation; the respective weights are determined by deleted interpolation.

Unknown words are handled by a suffix trie and successive abstraction. Average part-of-speech tagging accuracy reported for various languages is between 96% and 97%, which is at least as good as the state of the art results found in the literature. The accuracy for known tokens is significantly higher than unknown tokens. For German newspaper data, when the words seen before (the words in its lexicon) the results are 11% points better than for the words not seen before (97.7% vs. 86.6%). It should be mentioned that the accuracy for known tokens is high even with very small amounts of training data [9].

3.4. Persian Tree Tagger

PTT is based on decision-trees. A decision-tree assigns a class number for an input pattern using top-down filtering of pattern along tests (middle nodes of tree) [10]. In reality decision-trees are trees for indicating "classification rules" in classification of objects of a certain domain to a set of classes, that this definition in context of POS tagging is assigning tags to input words.

In decision-tree technique we deal with building blocks called ambiguity classes. The words are classified in these classes with regard to their possible tags. An ambiguity class can be consists of 2, 3 and more tags. For example a word that is

noun as well as verb in a corpus goes to noun-verb ambiguity class and every word resident in an ambiguity class is called an example for that class.

Context modeling is an important factor in decision-tree technique for accuracy of tagger. For example L. Marquez in [10] uses the below model for tagger:

$$\text{Tag}(-3) \text{ Tag}(-2) \text{ Tag}(-1) \text{ word} \text{ Tag}(+1) \text{ Tag}(+2)$$

Which means for assigning tag to word, we look at tags of 3 previous words and two next words of the word. But in MFT tagger we work context-independent and our induction is only based on ambiguity classes. Resulted accuracy using this model can be a baseline for accuracy of decision-tree technique.

We deal with statistical decision-trees instead of general decision-trees in context of POS tagging i.e. we must store number of examples belong to every class (tag) in tree nodes (ambiguity classes). This information will be used later for computing conditional probabilities.

4. OUR EXPERIMENTAL PROCESS

In order to do the experiments, some steps must be followed. These steps include preparing the corpus files for the three POS tagging systems (MLE, PTT and TnT), preparing test and training sets from the corpus and finally tagging the files by these systems. In this section, we will describe these steps.

4.1. Preparing the corpus

The untagged input files for each of the systems should have only one column of tokens of the text. If the line contains a space, all characters after the first space character are ignored. The format of tagged files required for each training set has only two columns with the same order as our corpus; it is similar to that of the untagged files but it extends the format by a second column: the first column is the token, and the second column is the tag. Everything after the second column is ignored.

The token in training and test files occupies all characters from the beginning of the line up to the first space and must not contain spaces. As some tokens in Persian have some spaces between their characters such as “بر می گردم” or ”BAR MI GARDAM”, a conversion program is implemented to remove these spaces from the tokens. It is clear that removing these spaces does not affect the accuracy of systems.

Table 3. The tags distribution

Tag Name	Frequency in Corpus	Probability
ADJ	22	8.46826E-06
ADJ_CMPR	7443	0.002864966
ADJ_INO	27196	0.010468306
ADJ_ORD	6592	0.002537398
ADJ_SIM	231151	0.088974829
ADJ_SUP	7343	0.002826473

ADV	1515	0.000583155
ADV_EXM	3191	0.001228282
ADV_I	2094	0.000806024
ADV_NEGG	1668	0.000642048
ADV_NI	21900	0.008429766
ADV_TIME	8427	0.003243728
AR	3493	0.001344528
CON	210292	0.080945766
DEFAULT	80	3.07937E-05
DELM	256595	0.098768754
DET	45898	0.017667095
IF	3122	0.001201723
INT	113	4.34961E-05
MORP	3027	0.001165155
MQUA	361	0.000138956
MS	261	0.000100464
N_PL	160419	0.061748611
N_SING	967546	0.372428585
NN	2	7.69842E-07
NP	52	2.00159E-05
OH	283	0.000108933
OHH	20	7.69842E-06
P	319858	0.123119999
PP	880	0.00033873
PRO	61859	0.023810816
PS	333	0.000128179
QUA	15418	0.005934709
SPEC	3809	0.001466163
V_AUX	15870	0.006108693
V_IMP	1157	0.000445353
V_PA	80594	0.031022307
V_PRE	42495	0.01635721
V_PRS	51738	0.019915033
V_SUB	33820	0.013018022
Max	967546	0.372428585
Min	2	7.69842E-07
Sum	2597937	1

4.2. Providing Test and Training Sets

In the majority of the part of speech tagging approaches, the sample is often subdivided into "training" and "test" sets. The training set is generally used for learning, i.e. fitting the parameters of the tagging systems. It also used for training of our method. The test set is for assessing the performance of the tagging systems. Again same test set used for testing of our method.

In our experiments, we split the corpus into to part 80% for training set and 20% for testing set. In addition, we added an identification number to each token of training and

test sets that is unique in whole of the collection. This number is used to identify what tag was assigned to each word in the collection by the three systems. We provided the training and test sets to all the three systems for the training and test steps.

4.3.The Training and Test Process

In the first step of this process, we trained all the three tagging systems separately by using the same training set for all of them that was prepared in the previous step. Then we gave each word of the test set to the MLE, PTT and TnT tagging systems and after gathering their results, we used them to construct the dataset matrix that contains real tags in the corpus and the tags that the three systems assigned to each token. We used this matrix to build confusion matrix. After this step, our methods like OWA, Max (Max ()), Max (Avg ()) and Predefined weighting, produce their results and assign a tag to each word. Then we compare the produced results and the three tagging systems results with real ones in the test set. Then we calculated the accuracy of each of them.

5. EXPERIMENTAL RESULTS

We implemented our method for a corpus with the size of 2598215 words. The size of training set was 2078595 words(80%) and the size of test set was 519620 words(20%) as described before.

In OWA method, we use some defined coefficients for each sorted column of confusion matrix. Coefficients are in this format:

$$\alpha^n \times (1 - \alpha), \alpha^{n-1} \times (1 - \alpha)^1, \alpha^{n-2} \times (1 - \alpha)^2, \dots$$

In first step we considered $\alpha = 0.5$ and ran our OWA fusion method and the results are shown and compared with the other tagging systems, in table 4. We should stated that the authors of MLE, PTT and TnT systems reported better performance than what is shown here and that's because of the larger portion of corpus that they used in their experiments (They chose training 85% and test 15%).

Table 4. Comparison of the results

Tagging system or method	precision
MLE	91.77
PTT	91.20
TNT	95.15
OWA	94.66
MaxMax	94.08
MaxAvg	94.66
Predefined weighting	94.67

As it can be seen, OWA method could not over perform TnT method. So, in the second step, we applied MaxMax, MaxAvg and Predefined weighting methods. It should be mentioned that in MaxMax method we use maximum value of each column of confusion matrix. But in MaxAvg method we use average of each column of confusion matrix and in Predefined weighting method we use some predefined coefficient for each sorted column of confusion matrix (we chose 0.35, 0.33 and 0.32 for TnT, MLE and PTT respectively). Predefined coefficients are determined based on accuracy of each of the three tagging systems. The results of MaxMax, MaxAvg and Predefined weighting are also shown in table 4.

In the third step of our experimentation investigated α parameter in OWA to see if it can improve our fusion results. So, we chose different values for the α parameter of OWA method and we came to the results that are shown in table 5.

Table 5. OWA results for different α

Alpha	precision
1	94.09
0.9	94.38
0.8	94.66
0.7	94.67
0.6	94.66
0.5	94.66
0.4	94.66
0.3	94.66
0.2	94.61
0.1	94.60
0.0	1.28

6. CONCLUSION AND FUTURE WORKS

In this paper we presented the OWA method to fuse the result of three different POS tagging systems. Also we presented our OWA experiment results with different α parameters. In addition to OWA method we applied MaxMax, MaxAvg and Predefined weighting.

Our results show that although OWA fusion technique has better results than MLE and PTT systems but it can not over perform TnT system. Now we are analyzing different parts of our model and our implementation to see why such a result happened. But one reason for this event may be the good performance of TnT tagger. TnT tagger could have learnt the characteristics of the other two systems in the training process, so fusion of the three systems could not improve the performance. So, we should investigate more to understand why OWA could not over perform TnT.

We think in addition to the train and test of 80% and 20%, we should investigate train and test sets of 85% and 15%, because it can produce more accuracy.

Another point that should be mentioned is the possibility of improving the performance of our fusion system over unknown words. We think as each of the three system act

differently toward unknown words, OWA fusion technique can have better performance than other systems, if we consider unknown words.

An evaluation of a statistical part of speech tagger known as TnT on Persian has been presented. In this work, a test collection for POS tagging was produced by reducing the tag set of a manually tagged corpus. The experiments were repeated several times in which the training and test sets were selected randomly from 85% and 15% of the collection respectively. The results show that the overall accuracy of the tagger is about 96.59% and the accuracy for known words is much higher than unknown words (about 24%).

As OWA is one of possible weighting methods, in future development of this study we are also to investigate other weighting methods.

ACKNOWLEDGEMENTS

Many thanks go to Mr. Hadi Amiri for his attention and help in our project. Also we appreciate Mr. M.Keikha and Mis F.Raja for giving us their project.

REFERENCES

- [1] F. Oroumchian, S. Tasharofi, H. Amiri, H. Hojjat, and F. Raja, "Creating a Feasible Corpus for Persian POS Tagging," *Technical Report*, no. TR3/06, University of Wollongong (Dubai Campus), May 2006.
- [2] Hadi Amiri, Mehdi Sarmadi, Hossein Hojjat, Farhad Oroumchian, "Memory based Part of Speech tagging Experiments with Persian Text", 2006.
- [3] Samira Tasharofi, Fahimeh Raja, Farhad Oroumchiana, Masoud Rahgozara, "Evaluation of statistical part of speech tagging of persian text", In International Symposium on signal processing and it's applications, February 2007.
- [4] M.Keikha, F.Mahdikhani, F.Oroumchian, "A decision tree based part-of-speech tagger for farsi language", 2007.
- [5] Ronald R. Yager, On ordered weighted averaging aggregation operators in multi-criteria decision making, *IEEE Transactions on Systems, Man and Cybernetics* 18, pp. 183-190, 1988.
- [6] M. BIJANKHAN, "The Role of the Corpus in Writing a Grammar: An Introduction to a Software," *Iranian Journal of Linguistics*, vol. 19, no. 2, fall and winter 2004.
- [7] Allen, J., "Natural Language Understanding", Second Edition. The Benjain/Cummings Publishing Company, Inc., Redwood City, California, USA, 1995.
- [8] R.L. Rivest, "Learning Decision Lists," *Machine Learning Journal*, vol. 2, no. 3, pp. 229-246, 1987.
- [9] T. Brants, "TnT – a Statistical Part-of-Speech Tagger," in Proc. sixth conference on applied natural language processing (ANLP-2000), Seattle, WA, 2000.
- [10] L. Marquez, "Part -of -Speech Tagging: A Machine -Learning Approach based on Decision Trees", PhD. Thesis, Dep. Llenguatges i Sistemes Informatics. Universitat Politecnica de Catalunya, 1999.